Work In Progress: Why Rely on Blind AIMDs?

Ioannis Psaras, Mehrdad Dianati, and Rahim Tafazolli

Center for Communication Systems Research (CCSR)
Dept. of El. Eng., University of Surrey
Guildford, GU2 7XH, Surrey, UK
(i.psaras, m.dianati, r.tafazolli)@surrey.ac.uk

Abstract. We are motivated by the fact that fixed Increase rates and Decrease ratios for AIMD cannot adjust TCP's performance to the Internet's diverse networking conditions. Indeed, we find that fixed values for the increase/decrease factors of AIMD restrain flexibility, which is a fundamental property of transport protocols in order to guarantee utilization and fairness in Modern and Future internetworks. We propose a new paradigm for hybrid AIMD designs that has the potential to adjust TCP's behavior according to network conditions.
The proposed *Multi-Rate AIMD* (MR-AIMD) increases additively the Additive Increase factor of AIMD in case of positive feedback and decreases multiplicatively (the AI factor) in case of negative feedback *and* Explicit Congestion Notifications. In other words, MR-AIMD takes into account ECN signals in order to quantify the level of network contention and adjusts its response accordingly.
We show that MR-AIMD reduces retransmission effort significantly, when contention is high, becomes aggressive when contention decreases and tolerates against random, transient errors due to fading channels.

Keywords: TCP, AIMD, Multi-Rate AIMD, Congestion Control, ECN

1 Introduction

The huge expansion of the Internet and the explosion of its applicability in our every day life has triggered extensive research in various fields of the networking technology. Clearly, TCP and its supporting AIMD algorithm is one of the most overworked topics during the last 15 years. Research efforts have focused on faster convergence to fairness [1] and efficiency [2], transmission over wireless lossy links [3], fast exploitation of high-speed links [4], [5], [6], [7], [8], [9] and optimization for web traffic [10], just to name a few.

Many researchers approached the above issues from a non-end-to-end point of view. Efforts on that direction focused on cooperation techniques between mobile hosts and base stations to improve TCP's performance over wireless media (e.g., [11]), or sophisticated AQM techniques to provide preferential treatment for short (web) flows (e.g., [12], [13], [14]).

Apart from their design-specific goals (i.e., tolerate wireless loss, converge to fairness, treat preferentially short flows), the above-mentioned approaches always target *full resource utilization*. Here, we argue that *full utilization does not* *necessarily translate into efficient utilization.* That is, when contention is low, the transport protocol should exploit all available resources. However, when contention increases, demand exceeds resource supply and therefore, full utilization should not be a matter of concern anymore. Instead, *efficient* utilization should become the challenge to deal with.

We consider that an efficient transport and congestion control mechanism should be aggressive when contention is low (in order to exploit available resources and tolerate against wireless errors) and conservative when contention increases (in order to reduce retransmission effort and decongest the buffers' queues). We argue that the fixed increase rates and decrease ratios for AIMD restrain flexibility and therefore, fail to provide *efficient* resource utilization. Motivated by similar studies such as TCP-SIMD [2], which however lacks the potential to tolerate against wireless errors and AIRA [15], [16], we attempt to design a hybrid congestion controller for future internetworks, which takes advantage of ECN signals. The Explicit Congestion Notification mechanism has been shown to provide some benefits for web traffic (e.g., [14], [17]). However, not many studies elaborated on the potential benefits that ECN can provide to long flows, or on its properties as an error discriminator.

We investigate the properties of a *Multi-Rate*, AIMD-based, Additive Increase (AI) factor. Briefly, the algorithm operates as follows: upon successful delivery of *cwnd* number of packets to the receiver side, not only the *cwnd*, but also the *the Additive Increase factor* increases Additively (i.e., $a \leftarrow a + \frac{a'}{cwnd}$), while on the face of loss, *the Additive Increase factor* is Multiplicatively Decreased (i.e., $a \leftarrow a - b \cdot a$). Decisions as to whether the AI factor should be increased or decreased and by how are based on AQM techniques, namely ECN.

The novelty of the proposed algorithm lies on its ability to adjust according to network conditions. MR-AIMD becomes aggressive when contention is low, although we explicitly note that it does not target high-speed environments; conservative when contention increases and tolerant against wireless errors, since it exploits ECN signals. We also note that although ECN is not famous for its capability as an error discriminator, our initial results show that there exists a lot of space for exploitation of such a system property.

2 Motivation: Blind AIMD

Deployment of AIMD is associated with two operational standards: (i) the fixed increase rate and decrease ratio and (ii) the corresponding selection of appropriate values.

Recent research has focused on altering the values for the *Additive Increase*, a, and *Multiplicative Decrease*, b, factors, in order to achieve fast bandwidth exploitation (e.g., [4], [5], [6], [7], [8], [9]) or faster convergence to fairness (e.g., [2], [1] and references therein), but has not questioned really the validity and efficiency of fixed rates throughout the lifetime of participating flows. In this context, research efforts cannot address questions such as: *Why do flows increase* *their rate by "a" packets instead of "2a" packets, even when half users of a system leave and bandwidth becomes available?*

2.1 Congested Wired Network

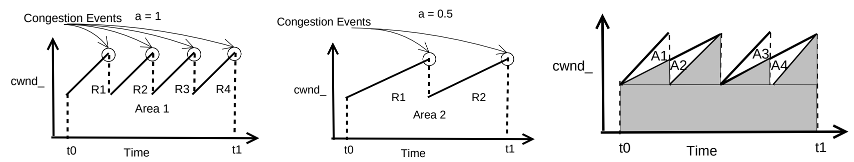
One possible justification for not highlighting the above research direction is that:

The Additive Increase factor of AIMD does not contribute to the long-term Goodput performance of TCP, when losses are due to buffer overflow.

In Figure 1, we present the *cwnd* evolution for two TCP flavors: Figure 1(a), where $a = 1$ (regular TCP) and Figure 1(b), where $a = 0.5$. The area underneath the solid *cwnd* lineplot (Area 1 and 2) represents the Goodput[1] performance of the protocols. In Figure 1(c), we show that both protocols achieve the same Goodput performance, since $A1 = A2$ and $A3 = A4$[2]. However,

Additive Increase affects significantly the Retransmission Effort of flows, which impacts overall system behavior as well.

For example, TCP $a = 1$, in Figure 1, experiences 4 congestion events, while TCP $a = 0.5$ experiences only 2. Assuming that each congestion event is associated with a fixed number of lost packets, regular TCP (i.e., $a = 1$) will retransmit twice as many packets as TCP with $a = 0.5$, without any gain in Goodput.



(a) Increase Factor = 1 (b) Increase Factor = 0.5 (c) Difference

Fig. 1. Different Increase Factors

We verify the above observations through simulations (using ns-2). We simulate TCP-SACK flows, for 200 seconds, over a single bottleneck dumbbell network topology (Figure 2); the backbone link transmits 1Mbps, its propagation delay is 20ms and the RED Router has buffer capacity equal to 25 packets.

[1] We define the system Goodput as $\frac{Original\ Data}{Connection\ Time}$, where *Original Data* is the number of bytes delivered to the high level protocol at the receiver (i.e., excluding retransmissions and the TCP header overhead) and *Connection Time* is the amount of time required for the data delivery. Instead, system Throughput includes retransmitted packets and header overhead (i.e., $\frac{Total\ Data}{Connection\ Time}$).

[2] In Figure 1(c) grey areas are common for both protocols; white areas are equal ($A1$ is similar to $A2$ and $A3$ is similar to $A4$).

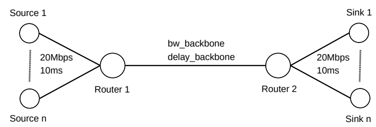


Fig. 2. Dumbbell Network Topology

Clearly, there is a tradeoff between *Aggressiveness* and *Retransmission Effort* (see Table 1). The degree of *Aggressiveness* that a transport protocol can achieve is tightly associated with its Retransmission Effort. The higher the Additive Increase factor, the more the retransmission effort of the transport protocol (see for example, the 4 flow, wired scenario in Table 1). Note that further increasing the level of contention may even degrade the system Goodput performance, due to timeout expirations [18], which are not considered in Figure 1.

In high contention scenarios, the higher the Additive Increase factor, the more retransmissions it causes, with zero gains in system Goodput.

Table 1. TCP Performance - Different Increase Factors

Wired	Goodput	Retransmissions		**Wireless**	Goodput	
	2/4 flows	2 flows	4 flows		2 flows	4 flows
				$a = 5$	114.8 KB/s	238.4 KB/s
$a = 2$	118.9 KB/s	742 pkts	1653 pkts	$a = 2$	81.3 KB/s	179.9 KB/s
$a = 1.5$	118.9 KB/s	548 pkts	1777 pkts	$a = 1.5$	73.5 KB/s	150.5 KB/s
$a = 1$	118.8 KB/s	278 pkts	704 pkts	$a = 1$	63.1 KB/s	127.9 KB/s
$a = 0.5$	118.9 KB/s	172 pkts	452 pkts	$a = 0.5$	44.2 KB/s	89 KB/s

2.2 Wireless, Mobile Computing

On the contrary, losses due to congestion may not always be the case. The evolution of mobile, wireless networking calls for further investigation and adjustment of transport layer algorithms to deal with losses due to wireless, fading channels as well. In this context,

The Additive Increase factor of AIMD may very well impact TCP's Goodput performance, when contention is low and losses are due to wireless errors.

We repeat the previous simulation to verify the above statement. The backbone link can now transfer 10Mbps (instead of 1Mbps) and we additionally insert 0.3 Packet Error Rate (PER) to emulate losses due to fading, wireless channels. The results are presented on the right side of Table 1. We observe that in case of low contention and transient errors due to fading channels, higher values for the Additive Increase factor of AIMD can boost TCP's performance significantly.

In low contention scenarios, where transient losses happen due to fading channels, the higher the Additive Increase factor, the more the Goodput gains for TCP.

2.3 Bandwidth Exploitation Properties

Today's Internet application and infrastructure heterogeneity demands for responsive transport protocols, which are able to exploit extra available bandwidth rapidly, in case of contention decrease; at the same time the transport layer protocol should be able to adjust its transmission rate downwards in case of incoming flows, in order to (i) leave space for the new flows and (ii) not overflow the network. That said, fixed Additive Increase provides fixed transmission rate acceleration both in case of contention decrease and in case of extra bandwidth constraints. We argue that such behavior is undesirable indeed, since it leads to slow bandwidth exploitation, when bandwidth becomes available, while it requires significant retransmission effort when bandwidth constraints prevail.

In case of contention decrease / increase scenarios, fixed acceleration leads to either slow resource exploitation or high retransmission effort, respectively.

3 MR-AIMD: Multi-Rate AIMD

3.1 The Algorithm

Regular TCP increases its congestion window by 1 packet, upon successful transmission of *cwnd* number of packets (i.e., $cwnd \leftarrow cwnd + \frac{a}{cwnd}$ on every ACK), while negative feedback (i.e., three duplicate ACKs), which is interpreted as network congestion, triggers *multiplicative cwnd* decrease (i.e., $cwnd \leftarrow cwnd - b \cdot cwnd$), where $a = 1$ and $b = 0.5$, according to [19].

As an initial approach to a "non-blind", dynamically adjustable increase factor, we attempt to graft the basic AIMD functionality *into the Additive Increase factor of TCP*. More precisely, the *Multi-Rate AIMD* algorithm increases the *cwnd* value according to:

$$cwnd \leftarrow cwnd + \frac{a \leftarrow a + \frac{a'}{cwnd}}{cwnd} \tag{1}$$

The initial value for a is 1, while for a' is 0.5. The proposed algorithm makes use of Active Queue Management (AQM) techniques in order to regulate the Additive Increase rate, a'. In particular, the algorithm uses the Explicit Congestion Notification (ECN) bit. There are two salient points that need to be clarified at this point regarding the cooperation scheme between the MR-AIMD sender and ECN: (i) an ECN marked packet triggers adjustment of the Additive Increase rate (a') *only* (i.e., the flow's *cwnd* is *not* reduced), and (ii) an ECN marked packet decelerates MR-AIMD's rate a' to 0.005 (instead of its initial value 0.5).

Modification of the ECN algorithm exhibits a number of desirable properties: (i) it smooths TCP's transmission rate and (ii) it avoids (to an extend)

TCP's drastic rate fluctuations, whenever deemed appropriate according to the proposed algorithm.

For the AI rate adaptation upon arrival of an ECN marked packet, we reason as follows: Once set, by the intermediate router, the ECN bit may trigger one of three possible responses: (i) Additive Increase, (ii) Multiplicative Decrease or (iii) stabilization *of the AI rate*, a'. The current implementation of MR-AIMD acts according to choice (i) (i.e., AI of rate a'). The rationale behind our choice is as follows: multiplicative decrease of the AI rate results in very low values for a' and therefore, conservative behavior. On the other hand, rate stabilization, through choice (iii), may result in system instability and flow unfairness, in the long term. Due to space limitations, we do not elaborate further on this issue, but we report that initial results verify our decisions for increased system performance (see Section 4).

The Additive Increase factor, a, decreases multiplicatively, according to Equation 2, upon a triple duplicate ACK event:

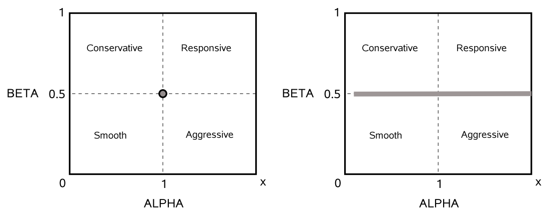
$$a \leftarrow a - b \cdot a. \tag{2}$$

The Multiplicative Decrease factor of MR-AIMD, b, is set to 0.5 similarly to TCP-AIMD (Equation 2), in order to guarantee fairness and stability [19]. Furthermore, upon a timeout event, MR-AIMD reduces the Additive Increase rate, a', to its initial value, 0.5, in order to account for increased levels of network contention. The rest of TCP's functions remain unchanged (e.g., RTO backoff, *cwnd* adjustments etc.). Although we do not elaborate on the convergence properties of MR-AIMD here, we report that according to our initial simulation results the algorithm indeed converges to fairness and stability, due (i) to its Multiplicative Decrease properties and (ii) to its ability to reduce retransmission effort, which implicitly increases system stability. We refer the reader to [16] for a more complete discussion on that topic.

3.2 Discussion

We assume that TCP's operational space, with regard to the Additive Increase and Multiplicative Decrease factors ALPHA and BETA, is represented by four basic domains: (i) conservative, (ii) aggressive, (iii) smooth and (iv) responsive (see Figure 3). The current, blind TCP-AIMD implementation covers a single point, only, within TCP's operational space (see Figure 3(a)). Clearly, the fixed increase/decrease parameters deal with none of the four operational domains, efficiency-wise and moreover, *any* pair of *fixed* increase/decrease parameters can deal with *one* operational domain *only*. We argue that such settings form an inflexible, conservative, worst-case approach to TCP's operational properties. For instance, a sophisticated transport layer algorithm should adjust according to network conditions: it should become conservative when contention is high, aggressive in case of transient wireless errors, responsive in case of contention increase/decrease and smooth in case of (relatively) static network load.

The proposed scheme extends TCP's functionality to operate along the x-axis of TCP's operational space. This way, several desirable properties are added to TCP's inherent functionality. For example, we show that careful design can lead to more aggressive transmission when contention is low and losses happen on the wireless portion of the network, while conservative transmission, when contention increases, can account for reduced retransmission overhead. The current proposal constitutes a first step on the further extension of TCP's functionality, in order to exploit the whole spectrum of possible behaviors (i.e., utilization of the y-axis as well). We note that although there have been some proposals on the same direction (e.g., [8], [20]), these proposals target high-speed environments and therefore have different design goals. Hence, we do not attempt to compare MR-AIMD with those approaches.



(a) TCP-AIMD (b) MR-AIMD

Fig. 3. x-AIMD Operational Space

We note that MR-AIMD does not target high-speed environments. Although the MR-AIMD's transmission rate may increase compared to regular TCP, its operational properties are not intended to exploit high-speed links. Instead, the proposed algorithm attempts to deal with the application diversity and infrastructure heterogeneity of present and future internetworks.

4 Preliminary Results

We use the SACK version of TCP with the timestamps option enabled. The simulation scenarios are similar to the ones presented in Section 2. That is, we use the dumbbell network topology[3], the queuing policy is RED and the buffer size is set according to the bandwidth-delay product of the outgoing link.

4.1 Congested Wired Networks

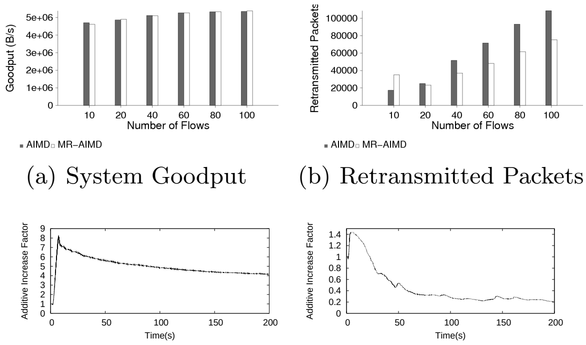
Initially, we simulate a wired network where the backbone link transfers 48 Mbps and induces propagation delay of 40ms. We repeat the simulation for increasing

[3] We have experimented with diverse-RTT topologies as well and we report that RED's inherent property to penalize higher-bandwidth flows, alleviates RTT-unfairness at least for RTT-differences in the order of 100ms or less. For the sake of simplicity and given the space limitations, we present results for equal-RTT flows only.

number of participating flows (from 10 to 100), to capture the performance of the proposed algorithm relatively with the level of contention.

We observe that when contention is low (e.g., 10 flows over 48Mbps) the proposed algorithm achieves the same Goodput performance as regular AIMD (see Figure 4(a)); the retransmission effort graph (Figure 4(b)) reveals that for low contention environments, the proposed AIMD operates aggressively. In Figure 4(c), we graph the average Additive Increase factor for a random flow, when 10 flows compete. This Figure verifies the aggressive behavior of MR-AIMD, when contention is low. We note that according to our solution framework this behavior is desirable indeed. That is, when contention is low we want the algorithm to be aggressive, ready to exploit extra bandwidth that may potentially become available due to flows that end their tasks and leave the system.

As contention increases, however, losses due to buffer overflow become more frequent, leading to multiplicative decreases of both the *cwnd* and the Additive Increase factor. In turn, smaller increase rates lead to reduced retransmission effort (see Figure 4(b)). In Figure 4(d), we present the average Additive Increase factor of MR-AIMD, for a random flow, when the total number of participating flows is 100. Indeed, we see that the average value of MR-AIMD's Additive Increase factor is below 1, allowing for less aggressive transmission, since the level of network contention so permits.



(a) System Goodput (b) Retransmitted Packets

(c) 10 Flows Experiment (d) 100 Flows Experiment

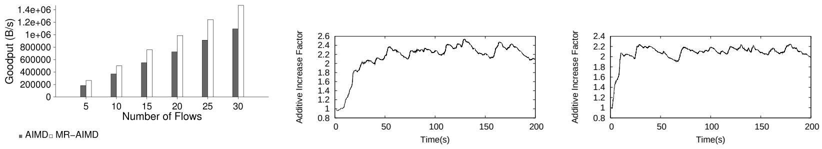
Fig. 4. Performance over Congested Wired Networks

Overall, we see that the proposed algorithm adjusts efficiently to the level of network contention, taking advantage of its dynamic increase/decrease acceleration properties to utilize resources accordingly. In particular, when contention is low the algorithm is aggressive, ready to utilize rapidly extra available bandwidth, while when contention increases the algorithm lowers the transmission rate to reduce retransmission effort.

4.2 Wireless Networks

We repeat the simulation presented in Section 2.2 to observe the performance of the proposed algorithm over lossy links. In the current setup the backbone link transfers 48Mbps with 40ms propagation delay, while the PER is 0.3.

Figure 5(a) depicts the outcome of the simulation. We see that the proposed algorithm is tolerant against random, transient errors caused by wireless, fading channels. MR-AIMD accelerates transmission faster than conventional AIMD, becoming more aggressive, when conditions permit, which is another desirable property in case of wireless errors [3]. Indeed, we see in Figures 5(b) and 5(c) that the Additive Increase factor is far above 1, allowing for speedy transmission and up to, approximately, 30% higher Goodput performance (Figure 5(a)) in case of errors due to wireless, lossy links.

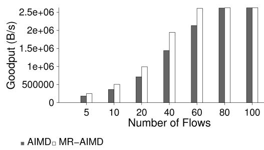


(a) System Goodput (b) 5 Flows Experiment (c) 30 Flows Experiment

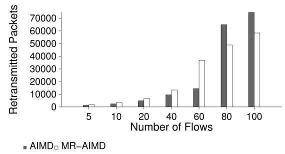
Fig. 5. Performance over Wireless Networks

4.3 Mixed Wired-Wireless Environments

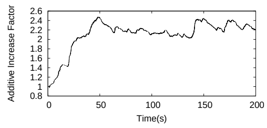
We perform one more experiment in order to verify that MR-AIMD adjusts correctly in mixed wired-wireless environments, where the level of contention may vary. The simulation environment is the same as previously, but the backbone link can now transfer 24Mbps. We repeat the simulation for increasing number of participating flows, from 5 to 100. Indeed, we see in Figure 6(a) that when contention is low MR-AIMD exploits the available resources, tolerates against random link errors and accelerates transmission (see Figure 6(c)), increasing the overall system Goodput (see Figure 6(a), flows 5-80). In contrast, when contention increases (i.e., 80 and 100 participating flows), MR-AIMD reduces its transmission rate, through lower Additive Increase factors (Figure 6(d)), although some errors may still be due to fading channels (i.e., we consider buffer overflow to be a more important factor for rate reduction than wireless errors). By doing so, MR-AIMD achieves the same Goodput performance as conventional AIMD, but reduces the retransmission effort of the transport protocol (Figure 6(b)). The present experiment verifies the hybrid behavior of MR-AIMD, which based on ECN signals adapts appropriately to the network conditions. Although ECN is not famous as an error discriminator, our initial results show that ECN-capable transports may be benefited, at least to an extend, from its operation as such. Further experimentation is needed in order to uncover ECN's capabilities regarding its accuracy on that direction.



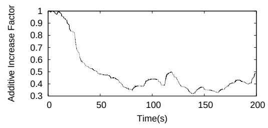
(a) System Goodput



(b) Retransmitted Packets



(c) 5 Flows Experiment



(d) 80 Flows Experiment

Fig. 6. Performance over Mixed Wired-Wireless Networks

4.4 Bandwidth Exploitation Properties

We attempt to briefly assess the bandwidth exploitation properties of the proposed algorithm. The Additive Increase factor of MR-AIMD progresses in time according to:

$$a_n = a_{n-1} + a' \cdot n, \ \ where \ n \geq 1. \tag{3}$$

In Equation 3, n stands for the number of RTTs, and a' is the acceleration factor of MR-AIMD (i.e., either 0.5 or 0.005). The initial value for a_n, for a new connection for example, is 1. Otherwise, for an existing connection, the initial value of a_n depends on the algorithm's state (i.e., AI through Equation 1, or MD through Equation 2).

In turn, MR-AIMD's *cwnd* after n RTTs is given by:

$$W_{fin} = W_{init} + \sum a_n, \tag{4}$$

TCP's *cwnd* after n RTTs is given by:

$$W_{fin} = W_{init} + a \cdot n, \tag{5}$$

where W_{init} is the initial *cwnd* and W_{fin} is the *cwnd* after n RTTs. Obviously, for TCP-AIMD $a = 1$, while for MR-AIMD a' is either equal to 0.5 or 0.005.

We assume a contention decrease event, where a number of participating flows leave the system when $W_{init} = 20$. From that point onwards, the rest of the flows have to exploit the extra bandwidth as fast as possible. We assume that since contention has decreased there are no ECN signals to the TCP sender (at least up to a certain point where contention becomes high due to the increased congestion windows of the rest of the participating flows).

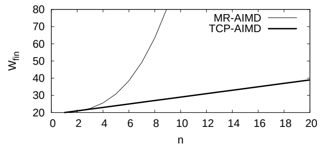


Fig. 7. Extra Bandwidth Exploitation Properties

As expected, we see in Figure 7 that MR-AIMD has the potential to exploit extra available network resources rapidly, without threatening the system's stability. That is, although initially the algorithm appears aggressive (MR-AIMD doubles its window within 6 RTTs, while TCP-AIMD needs 20 RTTs), it will immediately slow down, when ECN marked packets indicate incipient congestion. Due to limited space, we do not elaborate further on that issue here.

5 Conclusions

We argue that a blind Additive Increase factor for AIMD limits TCP's performance in terms of efficient resource utilization. We proposed a rather simple but novel approach towards a new design space for transport layer internetworking. Although the proposed settings are chosen based on experimental evaluations only, they seem to boost TCP's performance significantly. Moreover, additional modifications can easily be incorporated. For example, we did not evaluate here the properties of MR-AIMD with regard to the RTT-unfairness problem of TCP. Although one may argue that the proposed algorithm, in its current form, may extend TCP's inability to treat diverse RTT flows fairly, simple modifications can improve TCP's performance on that direction as well. For instance, MR-AIMD's Additive Increase function may be complemented with a fraction of the flow's measured RTT sample (i.e., $a \leftarrow a + \frac{c \cdot a'}{cwnd}$, where c is the flow's latest measured RTT sample). We note, however, that since MR-AIMD requires AQM techniques, namely RED with ECN to be implemented in the intermediate router [21], RTT-unfairness issues are partially eliminated due to RED's inherent properties, as our initial results (not included here) indicate.

6 ACKNOWLEDGMENTS

This work is supported in part by the European Commission through the *4WARD* project in the 7^{th} *Framework Programme*. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the *4WARD* project or the European Commission.

References

1. Lahanas, A., Tsaoussidis, V.: Exploiting the Efficiency and Fairness Potential of AIMD-based Congestion Avoidance and Control. Computer Networks **43**(2) (2003) 227–245
2. Jin, S., Guo, L., Matta, I., Bestavros, A.: TCP-friendly SIMD Congestion Control and Its Convergence Behavior. In: Proc. of 9th IEEE ICNP, Riverside, CA. (2001)
3. Tsaoussidis, V., Matta, I.: Open issues on TCP for Mobile Computing. Wiley, WCMC **2**(1) (February 2002)
4. Floyd, S.: HighSpeed TCP for Large Congestion Windows, RFC 3649 (2003)
5. Kelly, T.: Scalable TCP: Improving Performance in Highspeed Wide Area Networks. ACM SIGCOMM CCR **33**(2) (2003) 83–91
6. et al., D.X.W.: FAST TCP: motivation, architecture, algorithms, performance. IEEE/ACM Transactions on Networking (2007)
7. Rhee, I., Xu, L.: CUBIC: A New TCP-Friendly High-Speed TCP Variant. In: Proceedings of PFLDnet. (2005)
8. et al., K.T.: A Compound TCP Approach for High-Speed and Long Distance Networks. In: Proc. of INFOCOM 2006, Barcelona, Spain
9. King, R., Baraniuk, R., Riedi, R.: TCP-Africa: An Adaptive and Fair Rapid Increase Rule for Scalable TCP. In: Proceedings of INFOCOM 2005. Volume 3. (2005) 1838–1848
10. Le, L., Aikat, J., Jeffay, K., Smith, F.D.: The effects of active queue management on web performance. In: SIGCOMM '03. (2003) 265–276
11. Bakre, A., Badrinath, B.R.: I-TCP: Indirect TCP for mobile hosts. In: Proc. 15th Intl Conf. on Distributed Computing Systems. (1995) 136–143
12. Mahajan, R., Floyd, S., Wetherall, D.: Controlling high-bandwidth flows at the congested router. In: ICNP '01. (2001) 192–201
13. Guo, L., Matta, I.: The war between mice and elephants. In: Proceedings of ICNP. (October 2001)
14. Le, L., Aikat, J., Jeffay, K., Smith, F.D.: Differential congestion notification: Taming the elephants. In: ICNP '04. (2004) 118–128
15. Psaras, I., Tsaoussidis, V.: AIRA: Additive Increase Rate Accelerator. In: Proc. of IFIP Networking 2008, Singapore. (May 2008) 691–702
16. Psaras, I., Tsaoussidis, V.: On the Properties of an Additive Increase Rate Accelerator. Computer Networks, to appear
17. Kuzmanovic, A.: The power of explicit congestion notification. In: SIGCOMM '05. (2005) 61–72
18. Psaras, I., Tsaoussidis, V.: Why TCP timers (still) don't work well. Computer Networks **51**(8) (2007) 2033–2048
19. Jacobson, V.: Congestion avoidance and control. In: Proc. of SIGCOMM. (1988) 314–329
20. R.N.Shorten, D.J.Leith: H-TCP: TCP for high-speed and long-distance networks. In: Proc. PFLDnet, Argonne. (2004)
21. Medina, A., Allman, M., Floyd, S.: Measuring the evolution of transport protocols in the internet. SIGCOMM Comput. Commun. Rev. **35**(2) (2005) 37–52